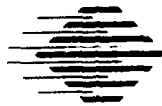
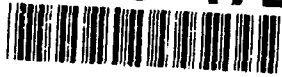


2



Carnegie-Mellon University  
Software Engineering Institute

AD-A253 172



## SAME Standard Package Installation Guide

Marc H. Graham  
May 1989

DTIC  
ELECTE  
JUL 27 1982  
S D

DISTRIBUTION STATEMENT A  
Approved for public release  
Distribution Unlimited

92 7 24 048

92-20040



The following statement of assurance is more than a statement required to comply with the federal law. This is a sincere statement by the university to assure that all people are included in the diversity which makes Carnegie Mellon an exciting place. Carnegie Mellon wishes to include people without regard to race, color, national origin, sex, handicap, religion, creed, ancestry, belief, age, veteran status or sexual orientation.

Carnegie Mellon University does not discriminate and Carnegie Mellon University is required not to discriminate in admissions and employment on the basis of race, color, national origin, sex or handicap in violation of Title VI of the Civil Rights Act of 1964, Title IX of the Educational Amendments of 1972 and Section 504 of the Rehabilitation Act of 1973 or other federal, state, or local laws or executive orders. In addition, Carnegie Mellon University does not discriminate in admissions and employment on the basis of religion, creed, ancestry, belief, age, veteran status or sexual orientation in violation of any federal, state, or local laws or executive orders. Inquiries concerning application of this policy should be directed to the Provost, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; telephone (412) 268-6684 or the Vice President for Enrollment, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; telephone (412) 268-2056.

**Special Report**

**CMU/SEI-89-SR-5**

**May 1989**

# **SAME Standard Package Installation Guide**



**Marc H. Graham**

Binding of Ada and SQL Project

*Approved for public release.  
Distribution unlimited.*

**Software Engineering Institute**  
Carnegie Mellon University  
Pittsburgh, Pennsylvania 15213

This technical report was prepared for the

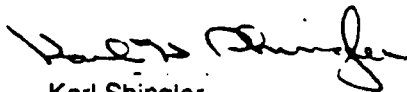
SEI Joint Program Office  
ESD/AVS  
Hanscom AFB, MA 01731

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

#### **Review and Approval**

This report has been reviewed and is approved for publication.

FOR THE COMMANDER



Karl Shingler  
SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

Copyright © 1989. Software Engineering Institute, Carnegie Mellon University

This document is available through the Defense Technical Information Center. DTIC provides access to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential contractors and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center, Attn: FDRA, Cameron Station, Alexandria, VA 22304-6145.

Copies of this document are also available through the National Technical Information Service. For information on ordering, please contact NTIS directly: National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

Use of any other trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

## Table of Contents

1. Introduction	1
2. Modifying the Source	1
2.1. SQL_Standard and SQL_System	1
2.2. SQL_Communications_Pkg and SQL_Database_Error_Pkg	2
2.3. To_String and To_SQL_Char_Not_Null	3
3. Compilation Order	4

**DTIC QUALITY INSPECTED 4**

<b>Accession For</b>	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
<b>Availability Codes</b>	
<b>Dist</b>	<b>Avail and/or Special</b>
A-1	



## List of Figures

<b>Figure 1:</b> The Package SQL_Standard	2
<b>Figure 2:</b> The Package SQL_System	3

## 1. Introduction

This document outlines the procedures for installing the SQL Ada Module Extensions (SAME) standard packages. It assumes that the source of these packages has been off-loaded from the distribution medium to a machine with the desired Ada compiler. (A description of the distribution medium accompanies the medium.)

Installing the SAME standard packages is a two-step operation. First, a subset of the source must be modified; second, the source is compiled into the proper Ada library or libraries.

## 2. Modifying the Source

There are six files on the distribution medium which need to be modified before compilation. These files are:

1. SQL\_Standard (a package specification)
2. SQL\_System (a package specification)
3. SQL\_Communications\_Pkg\_Body (a package body)
4. SQL\_Database\_Error\_Pkg\_Body (a package body)
5. To\_String (a subunit body)
6. To\_SQL\_Char\_Not\_Null (a subunit body)

### 2.1. SQL\_Standard and SQL\_System

The packages SQL\_Standard and SQL\_System (see Figures 1 and 2) contain type definitions and constants that describe the DBMS to the SAME. Strictly speaking, the package SQL\_Standard is part of the ANSI SQL standard. The type and subtype definitions in SQL\_Standard describe the standard ANSI database types INT, SMALLINT, REAL, DOUBLE PRECISION, and CHAR.

The information needed to describe the numeric types can be found in the documentation supplied by the DBMS vendor. If, for example, the DBMS data type INTEGER (or INT) references 32-bit integers, and SMALLINT references 16-bit integers, these type definitions should be:

```
type Int is range -(2**31) .. (2**31) - 1;
type Smallint is range -(2**15) .. (2**15) - 1;
```

The treatment of character data in SQL\_Standard is meant to support the character sets defined by the implementor. If the DBMS character set is ASCII, then the definitions in SQL\_Standard should read:

```
package Character_Set renames Standard;
subtype Character_Type is Character_Set.Character;
```

Otherwise, it is assumed there exists an Ada package that supports the DBMS character set for Ada programs. Assuming that that package is called Host\_Character\_Support\_Pkg, and that the character enumeration type definition within that package is named *Character*, the above definitions should read:



```

package Character_Set renames Host_Character_Support_Pkg;
subtype Character_Type is Character_Set.Character;

```

The package SQL\_System declares two constants whose values cannot be deduced from the information in SQL\_Standard. The constant MAXCHRLen is the length of the longest string the DBMS will store. This value must be in the range of the Ada predefined type Standard.Integer for the target compiler. The constant MAXERRLEN is the length of the longest error message the DBMS error message routine will produce. This error message routine or function, which is not controlled by the ANSI standard, varies from DBMS to DBMS. The value of MAXERRLEN must not exceed MAXCHRLen. Setting MAXERRLEN to MAXCHRLen will usually work, but may waste space.

**Note Well:** The packages SQL\_Standard and SQL\_System are basic to every SAME application. Recompilation of these packages necessitates recompilation of every SAME standard package and application. It is imperative that the values in these packages be correctly set.

```

package Sql_Standard is
package Character_Set renames csp;
  subtype Character_Type is Character_Set.cst;
  type Char is array (positive range <>)
    of Character_Type;
  type Smallint is range bs..ts;
  type Int is range bi..ti;
  type Real is digits dr;
  type Double_Precision is digits dd;
-- type Decimal is to be determined;
  type Sqlcode_Type is range bsc..tsc;
  subtype Sql_Error is Sqlcode_Type
    range Sqlcode_Type'FIRST .. -1;
  subtype Not_Found is Sqlcode_Type
    range 100..100;
  subtype Indicator_Type is t;

-- csp is an implementor-defined package and cst is an
-- implementor-defined character type. bs, ts, bi, ti, dr,
-- dd, bsc, and tsc are implementor-defined integral values.
-- t is int or smallint corresponding to an
-- implementor-defined <exact numeric type> of indicator
-- parameters.

end sql_standard;

```

Figure 1: The Package SQL\_Standard

## 2.2. SQL\_Communications\_Pkg and SQL\_Database\_Error\_Pkg

The packages SQL\_Communications\_Pkg and SQL\_Database\_Error\_Pkg are used for SAME standard error processing. (See the document *Guidelines for the Use of the SAME*, CMU/SEI-89-TR-16, for details.) The installation of these packages may require modifying their bodies. Modification of the specifications of these packages is not generally required. The body of SQL\_Communications\_Pkg will depend on the platform, that is, the Ada compiler and DBMS. The body of SQL\_Database\_Error\_Pkg is independent of the platform but dependent on decisions to be made at installation time.

```

-- SQL_System is a "platform-specific" package
--   within the SAME
package SQL_System is
  -- MAXCHRLen is the upper bound of the SQL_Char_Pkg
  --   subtypes SQL_Char_Length and SQL_Unpadded_Length
  -- SQL_Char_Length is a subtype of Natural with a lower
  --   bound of 1
  -- SQL_Unpadded_Length is a subtype of Natural with a lower
  --   bound of 0

  MAXCHRLen : constant integer := str_leng;

  -- MAXERRLEN is the maximum length of the error message
  --   string returned from the DBMS error message function

  MAXERRLEN : constant integer := msg_leng;

end SQL_System;

```

Figure 2: The Package SQL\_System

The package SQL\_Database\_Error\_Pkg declares the procedure Process\_Database\_Error. This procedure is called by abstract module procedures when an unexpected DBMS exceptional condition (non-zero SQLCODE value) has occurred. The procedure should take whatever action is mandated by local policy for error reporting. The exception SQL\_Database\_Error, defined in SQL\_Communications\_Pkg, will be raised upon termination of Process\_Database\_Error.

The distributed version of Process\_Database\_Error calls the function SQL\_Database\_Error\_Message in package SQL\_Communications\_Pkg, which returns an explanatory error message acquired from the DBMS. It then prints the message with text\_io.put. The installer should modify this body as appropriate for his or her installation.

### 2.3. To\_String and To\_SQL\_Char\_Not\_Null

The subunits To\_String and To\_SQL\_Char\_Not\_Null are used for converting character representations. If the DBMS character set is ASCII, the distributed versions of these subunits can be used without modification.

Assume that is not the case. Assume further the package Host\_Character\_Support\_Pkg mentioned earlier contains routines for converting between its string type, Host\_String, a one dimensional array over the type Host\_Character\_Support\_Pkg.Character, and the Ada predefined type Standard.String. Specifically, suppose that within the support package are to be found functions with the following declarations:

```

function To_String (Value : Host_String) return String;
function To_Host_String (Value : String) return Host_String;

```

In that case, these subunits could be recoded as follows:

```

with Host_Character_Support_Pkg;
separate (SQL_Char_Pkg)
function To_String (Value : SQL_Char_Not_Null)
    return String is
package HCSP renames Host_Character_Support_Pkg;
begin
    return (HCSP.To_String(HCSP.Host_String(Value)));
    -- notice explicit Ada type conversion
end To_String;

with Host_Character_Support_Pkg;
separate (SQL_Char_Pkg)
function To_SQL_Char_Not_Null (Value : String)
    return SQL_Char_Not_Null is
package HCSP renames Host_Character_Support_Pkg;
begin
    return (SQL_Char_Not_Null(HCSP.To_Host_String(Value)));
    -- notice inverse Ada type conversion
end To_SQL_Char_Not_Null;

```

**Note:** Unlike the package specifications SQL\_Standard and SQL\_System, the remaining user modifications are all to bodies. Whereas modifications to the package specifications must be made before any compilations can take place, the remaining modifications can be done later, and redone, without recompilation effects.

### 3. Compilation Order

The SAME standard packages are divided into groups. Packages within a group may be compiled in any order. All packages in one group must be compiled before compiling packages in the next group. Notice that the bodies of packages are distributed in separate files from those containing the specifications of packages. These files are named <package name>\_Body.

#### Group I.

1. SQL\_Standard
2. SQL\_System
3. SQL\_Exceptions

#### Group II.

1. SQL\_Boolean\_Pkg

#### Group III.

1. SQL\_Char\_Pkg
2. SQL\_Real\_Pkg
3. SQL\_Double\_Precision\_Pkg

#### **Group IV.**

1. SQL\_Communications\_Pkg
2. SQL\_Int\_Pkg
3. SQL\_Smallint\_Pkg
4. SQL\_Enumeration\_Pkg

#### **Group V.**

1. SQL\_Base\_Types\_Pkg
2. To\_SQL\_Char\_Not\_Null
3. To\_String
4. SQL\_Decimal\_Pkg

#### **Group VI.**

1. SQL\_Database\_Error\_Pkg

